

Advancing Matrix Completion by Modeling Extra Structures beyond Low-Rankness

Guangcan Liu

Department of Statistics and Biostatistics
 Department of Computer Science
 Rutgers University
 Piscataway, NJ 08854, USA
 guangcan.liu@rutgers.edu

Ping Li

Department of Statistics and Biostatistics
 Department of Computer Science
 Rutgers University
 Piscataway, NJ 08854, USA
 pingli@stat.rutgers.edu

Abstract

A well-known method for completing low-rank matrices based on convex optimization has been established by Candès and Recht [1]. Although theoretically complete, the method may not entirely solve the low-rank matrix completion problem. This is because the method captures only the low-rankness property which gives merely a constraint that the data points locate on some low-dimensional subspace, but generally ignores the extra structures which specify in more detail how the data points locate on the subspace. Whenever the geometric distribution of the data points is not uniform, the coherence parameters of data might be large and, accordingly, the method might fail even if the latent matrix to recover is fairly low-rank. To better handle non-uniform data, in this paper we propose a model termed Low-Rank Factor Decomposition (LRFD), which imposes an additional restriction that the data points must be represented as linear combinations of the bases in a given dictionary. We show that LRFD can well handle non-uniform data, provided that the dictionary is configured properly: We mathematically prove that if the dictionary itself is low-rank then LRFD is immune to the coherence parameters which might be large on non-uniform data. This provides an elementary principle for learning the dictionary in LRFD and, naturally, leads to a practical algorithm for advancing matrix completion. Extensive experiments on randomly generated matrices and motion datasets show encouraging results.

1 Introduction

In modern applications such as *structure from motion*, very often one needs to restore the missing entries of a matrix, i.e., *matrix completion* [2]. In general, given no presumptions about the nature of the entries, matrix completion is virtually impossible as the missing entries can be of arbitrary values. Due to the low-rankness nature of today’s high-dimensional data, a commonly adopted assumption is that the latent matrix we want to recover is fairly low-rank, resulting in the so-called *low-rank matrix completion* problem, which is formulated as follows:

Problem 1 (Low-Rank Matrix Completion) *Suppose we have a data matrix $X \in \mathbb{R}^{m \times n}$, which is known only on a fraction of its entries:*

$$[X]_{ij} = [L_0]_{ij}, \forall (i, j) \in \Omega,$$

where $L_0 \in \mathbb{R}^{m \times n}$ is a low-rank matrix each column of which is a data point lying on some low-dimensional subspace, $[\cdot]_{ij}$ denotes the (i, j) th entry of a matrix, and $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$ is an index set consisting of the locations of the observed entries. Given the incomplete matrix X (and the index set Ω), can we exactly recover the latent matrix L_0 in a scalable way?

There is a large community that explores the above problem using various statistical tools, e.g., [1, 3, 4, 5, 6, 8, 14, 22, 23, 24]. Of all those notable contributions, the most fundamental and significant one is probably the convex optimization based method established by Candès and Recht [1]. For the ease of presentation, we shall call this method as “CONO” (CONvex Optimization) for short. CONO tells us for sure that, when the low-rank matrix L_0 is meanwhile *incoherent* (i.e., with low coherence parameters), L_0 can be exactly recovered by using the following convex, parameter-free, and potentially scalable program:

$$\min_L \|L\|_*, \quad \text{s.t.} \quad \mathcal{P}_\Omega(X - L) = 0, \quad (1)$$

where $\|\cdot\|_*$ is the *nuclear norm* [9, 10] of a matrix, i.e., the sum of the singular values of a matrix, and \mathcal{P}_Ω denotes the orthogonal projection onto the linear space of matrices supported on Ω . Besides of its completeness in theory, CONO also has good empirical performance and is therefore widely regarded as a milestone in the history of matrix completion.

Nevertheless, CONO cannot be the best solution to the low-rank matrix completion Problem 1. Indeed, the method might be unsuccessful even when the latent matrix L_0 is strictly low-rank and the locations of missing entries are selected uniformly at random. This is because CONO captures only the low-rankness property of L_0 , but essentially ignores the *extra structures* which are critical to the success of recovery: Given the low-rankness constraint that the data points (i.e., columns vectors of L_0) locate on a low-dimensional subspace, it is quite normal that the data may have some extra structures which specify in more detail *how* the data points locate on the subspace, as illustrated in Figure 1. Notice that the extra structures are essentially nonlinear and hard to parameterize. Therefore, we shall not adopt parametric models to describe and explore each extra structure in a particular way, but instead generally divide all cases shown in Figure 1 into two categories:

- 1) **Uniform data:** The data points *uniformly* locate on a low-dimensional subspace, as shown in Figure 1(a).
- 2) **Non-uniform data:** The data points *non-uniformly* locate on a low-dimensional subspace, as shown in Figure 1(b) ~ Figure 1(e).

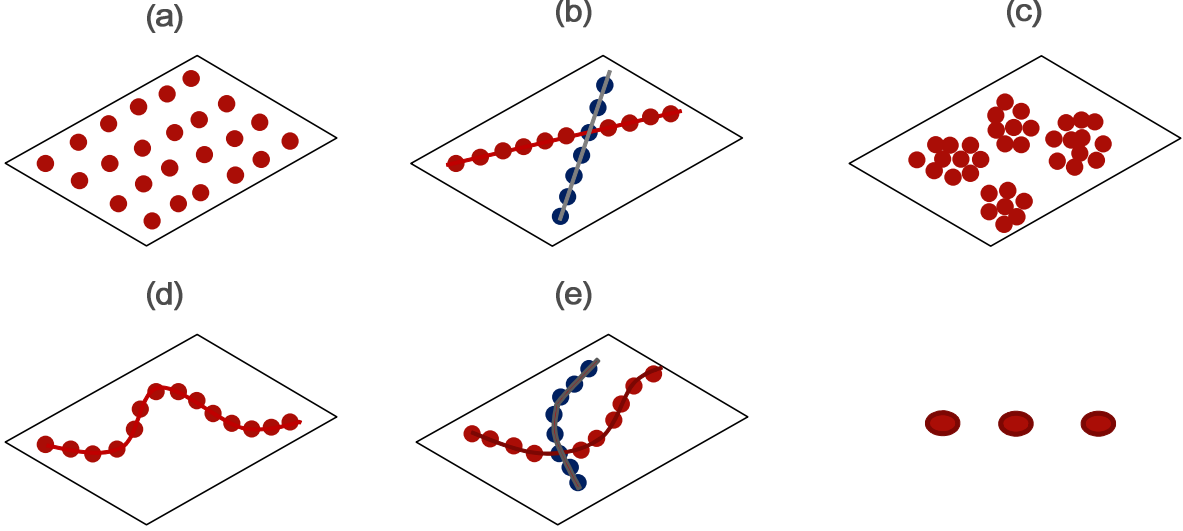


Figure 1: Illustrating the extra structures beyond low-rankness. Each column of the data matrix L_0 is a data point. Given the constraint that L_0 is low-rank, i.e., the data points locate on a low-dimensional subspace, more specific situation could be: (a) The data points uniformly distribute on the subspace, (b) the data points have a mixture structure of multiple “small” subspaces inside the “large” subspace, (c) the data points form multiple “ball-like” clusters, (d) the data points lie on a nonlinear manifold inside the subspace, (e) the data points follow a mixture structure of multiple nonlinear manifolds inside the subspace, etc.

For the uniform case as in Figure 1(a), CONO is probably the best method for low-rank matrix completion. Nevertheless, uniform data actually seldom exist in reality and CONO might not work well on non-uniform data. The reason is that the coherence parameters of non-uniform data might be large, and thus CONO might fail to recover L_0 even when L_0 is fairly low-rank. Even more, non-uniform data are ubiquitous in realistic areas such as computer vision. For example, it is known that the data matrix of trajectories of motion objects provably follows a mixture structure of multiple subspaces as in Figure 1(b) [11]. Anyway, uniform data is after all a special case of non-uniform data, and thus it is undoubtedly significant to study the matrix completion problem in the context of non-uniform data.

To accomplish an advanced solution to the low-rank matrix completion Problem 1 in the context of non-uniform data, in this paper we propose to consider a generalized version of (1), called as *Low-Rank Factor Decomposition* (LRFD) for the convenience of citation:

$$\min_Z \|Z\|_*, \quad \text{s.t.} \quad \mathcal{P}_\Omega(X - AZ) = 0, \quad (2)$$

where $A \in \mathbb{R}^{m \times d}$ is a dictionary matrix constructed or learnt in advance (the choice of the dictionary size d is immaterial). Note here that, unlike in CONO, in our LRFD it is AZ^* that reconstructs L_0 (assume Z^* is the minimizer to (2)). It is easy to see that (2) falls back to (1) when $A = \mathbf{I}$ (identity matrix). So it could be regarded that LRFD is a generalization of CONO.

To well handle non-uniform data, the dictionary matrix A should be chosen properly. We shall mathematically prove that if the dictionary itself is low-rank then LRFD is immune to the coherence parameters which might be large on non-uniform data. This provides an elementary principle for learning the dictionary in LRFD. Subsequently, we devise a practical algorithm to

obtain proper dictionaries in unsupervised environments. Our extensive experiments on randomly generated matrices and motion datasets show encouraging results. In summary, our contributions include:

- We propose to improve low-rank matrix completion by modeling the extra structures possibly existing in data. To our knowledge, we are the first to pursue this direction in the community of matrix completion. Furthermore, we establish a generic model termed LRFD, some elementary theories and a practical algorithm for resolving the problem of restoring a low-rank (yet non-uniform) matrix from its incomplete versions.
- The idea of replacing a variable L with the product of two variables, saying AZ , is essentially the spirit of *matrix factorization* which has been discussed for long, e.g., [7, 13, 14, 15, 25]. In that sense, the investigations of this paper help to understand why the factorization techniques could be effectual.
- While the concept of *coherence* is now standard and widely used in various literatures, e.g., [16, 17], there is a lack of studies about the *physical* regime that affects the behaviors of coherence parameters. This paper shows that the coherence parameters are related in nature to the geometric distribution of data points: The more non-uniformly the data points distribute, the larger the coherence parameters could be.

2 Summary of Main Notations

Capital letters such as M are used to represent matrices, and accordingly, $[M]_{ij}$ denotes its (i, j) th entry. The particular symbol $(\cdot)^+$ denotes the Moore-Penrose pseudo-inverse of a matrix, i.e., $M^+ = V_M \Sigma_M^{-1} U_M^T$ for any matrix M with SVD¹ $U_M \Sigma_M V_M^T$. Letters U , V , Ω and their variants (complements, subscripts, etc.) are reserved for column space, row space and index set, respectively. We shall abuse the notation U to denote the linear space spanned by the columns of U . The projection onto the column space, U , is denoted by \mathcal{P}_U and given by $\mathcal{P}_U(M) = U U^T M$. We shall also abuse the notation Ω to denote the linear space of matrices supported on Ω , and use \mathcal{P}_Ω and $\mathcal{P}_{\Omega^\perp}$ to respectively denote the projections onto Ω and Ω^c (i.e., the complement of Ω) such that $\mathcal{P}_\Omega + \mathcal{P}_{\Omega^\perp} = \mathcal{I}$, where \mathcal{I} is the identity operator.

Three types of matrix norms are used in this paper, and they are all functions of the singular values: 1) the operator norm or 2-norm (i.e., the largest singular value) denoted by $\|M\|$, 2) the Frobenius norm (i.e., the square root of the sum of squared singular values) denoted by $\|M\|_F$ and 3) the nuclear norm or trace norm (i.e., the sum of singular values) denoted by $\|M\|_*$. The only used vector norm is the ℓ_2 norm, which is denoted by $\|\cdot\|_2$.

The letter μ and its variants are reserved to denote the coherence parameters of a matrix. We also reserve two lowercase letters, m and n , to respectively denote the data dimension and the number of data points, and we use the following two symbols throughout this paper:

$$n_1 = \max(m, n) \quad \text{and} \quad n_2 = \min(m, n).$$

3 Analysis, Theory and Algorithm

In this section, we shall try to answer the following two questions: (1) Why CONO might not work well on non-uniform data? (2) How to choose the dictionary matrix A in LRFD?

¹In this paper, SVD always refers to skinny SVD. For a rank- r matrix $M \in \mathbb{R}^{p \times q}$, its SVD is of the form $U_M \Sigma_M V_M^T$, where $U_M \in \mathbb{R}^{p \times r}$, $\Sigma_M \in \mathbb{R}^{r \times r}$ and $V_M \in \mathbb{R}^{q \times r}$.

3.1 Why CONO Might Fail on Non-Uniform Data?

To get a definite answer to the question highlighted above, we introduce below the concept of *coherence* and investigate the physical regime that affects the behaviors of coherence parameters.

The definition of coherence adopted by this paper is the same as [1, 16]. For a matrix $M \in \mathbb{R}^{p \times q}$ with rank r and SVD $U_M \Sigma_M V_M^T$, there are two coherence parameters, μ_1 and μ_2 , which are useful to characterize the statistical properties of the matrix. The first coherence parameter, $1 \leq \mu_1 \leq p$, which captures the statistical properties of the *column space* identified by U_M , is defined as

$$\mu_1(M) = \frac{p}{r} \max_i \|U_M^T e_i\|_2^2, \quad (3)$$

where e_i denotes the i th standard basis. The second coherence parameter, $1 \leq \mu_2 \leq q$, which characterizes the *row space* identified by V_M , is defined as

$$\mu_2(M) = \frac{q}{r} \max_i \|V_M^T e_i\|_2^2. \quad (4)$$

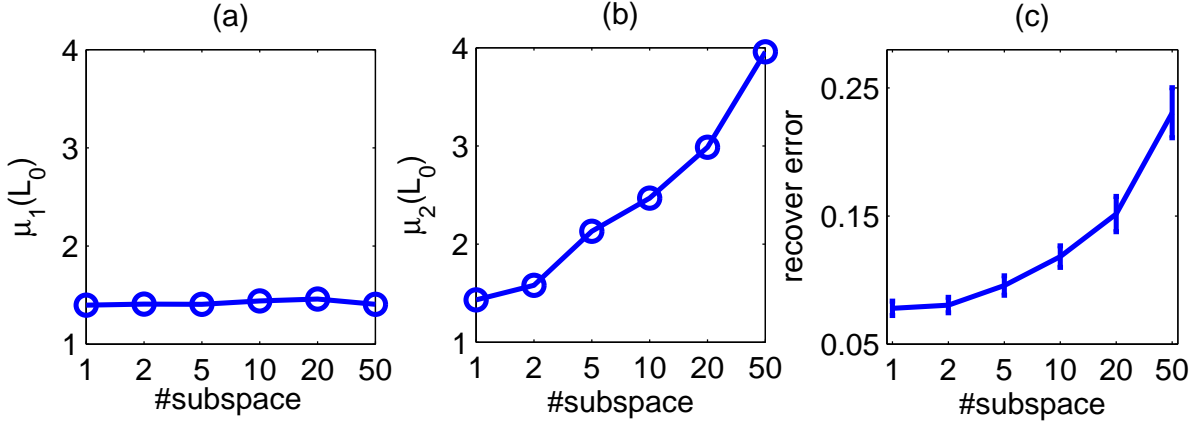


Figure 2: Exploring the properties of coherence parameters, using randomly generated matrices. The size of L_0 is fixed to be 500×500 . The underlying subspace number k varies from 1 to 50. We set the dimension of each subspace as $100/k$, and thus L_0 has a fixed rank of 100. (a) The first coherence parameter $\mu_1(L_0)$ vs subspace number. (b) The second coherence parameter $\mu_2(L_0)$ vs subspace number. (c) The performance of CONO vs subspace number. For the matrix completion experiments in (c), the percentage of missing entries is fixed to be 45%. The recover error is computed as $\|\hat{L}_0 - L_0\|_F / \|L_0\|_F$, where \hat{L}_0 is an estimate of L_0 . The numbers shown in (c) are collected from 100 random trials.

Since the behaviors of data points could affect the row space V_M , the second coherence parameter μ_2 may somehow depend on the geometric distributions of the data points. To confirm, we consider for exploration the mixture structure shown in Figure 1(b), which is about the phenomenon that the data points in L_0 are sampled from k number of subspaces, i.e., $L_0 = [L_0^{(1)}, \dots, L_0^{(k)}]$, where $L_0^{(i)}$ is the matrix of data points from the i th subspace. While the rank of L_0 is fixed and the underlying subspace number k goes large, Figure 2(b) shows that the second coherence parameter $\mu_2(L_0)$ keeps increasing. To see why the second coherence parameter increases with the cluster number underlying L_0 , please refer to [26].

Among other things, the information revealed by Figure 2(a) is remarkable and useful: The first coherence parameter $\mu_1(L_0)$ is immune to the variation of the underlying subspace number.

This is actually natural, because the behaviors of the data points can only affect the row space, while μ_1 is defined on the column space². Analogously, we have the following doctrines that depict the coherence parameters in general:

- The first coherence parameter $\mu_1(L_0)$ is always small, in despite of whether or not the geometric distribution of the data points is uniform.
- The second coherence parameter $\mu_2(L_0)$ is small on the uniform data, but could be large on the non-uniform cases such as Figure 1(b).

Now the answer to the question highlighted in the beginning of this subsection is clear. Namely, the analysis in [1] illustrates that CONO prefers the cases where both μ_1 and μ_2 are small. Nevertheless, such an expectation might not be true as the second coherence parameter μ_2 could be large on non-uniform data and, accordingly, the recovery performance of CONO might be unsatisfactory even when L_0 is strictly low-rank. To verify this assertion, we have executed lots of numerical experiments. As we can see from Figure 2(c), CONO degrades with the enlargement of the subspace number underlying L_0 , i.e., CONO is dropping while $\mu_2(L_0)$ is increasing. This phenomenon additionally reflects that, besides of the low-rankness property, the extra structures (beyond low-rankness) also have a dramatic influence on the recovery of the latent matrix L_0 .

3.2 How to Choose the Dictionary in LRFD?

As aforementioned, the first coherence parameter μ_1 is invariant to the variations of the geometric distribution of data points. Hence, a promising direction for recovering non-uniform data might be to figure out in which conditions LRFD can *avoid* the influences of the second coherence parameter μ_2 . We shall show that, when the dictionary A itself is low-rank, LRFD is able to get around of μ_2 . Namely, the following two theorems are proved without using μ_2 (The detailed procedures of proof can be found in Section 6).

Theorem 1 (Noiseless) *Let $U_0 \Sigma_0 V_0^T$ be the SVD of L_0 . Suppose that the dictionary matrix A with SVD $U_A \Sigma_A V_A^T$ satisfies $\mathcal{P}_{U_A}(U_0) = U_0$ (i.e., U_0 is a subspace of U_A). For any $\delta > 0$ and some numerical constant $c_a > 0$, if*

$$\text{rank}(L_0) \leq \text{rank}(A) \leq \frac{\delta^2 n_2}{c_a \mu_1(A) \log n_1} \text{ and } \frac{|\Omega|}{mn} \geq \delta,$$

then with probability at least $1 - n_1^{-10}$, the optimal solution (denoted as Z^) to problem (2) is unique and exact, in a sense that $Z^* = A^+ L_0$.*

Figure 3 further confirms that there exist some kind of dictionaries using which LRFD is immune to the second coherence parameter μ_2 . The condition $\mathcal{P}_{U_A}(U_0) = U_0$ (i.e., U_0 is a subspace of U_A) is indispensable if we ask for the exactness of recovery, as $U_0 \subset U_A$ is implied by the equality $AZ^* = L_0$. So what is suggested by above theorem is that the dictionary matrix A should be made low-rank. This provides an elementary criterion for learning the dictionary matrix of LRFD.

The program (1) is designed for the case where the observed entries are noiseless. In reality this is often not true and the observations themselves could be actually contaminated. Candès and Plan

²Notice that μ_1 could be also large if the row vectors of L_0 own some structures beyond low-rankness. Such kind of data exist widely in text domain and we leave this as future work.

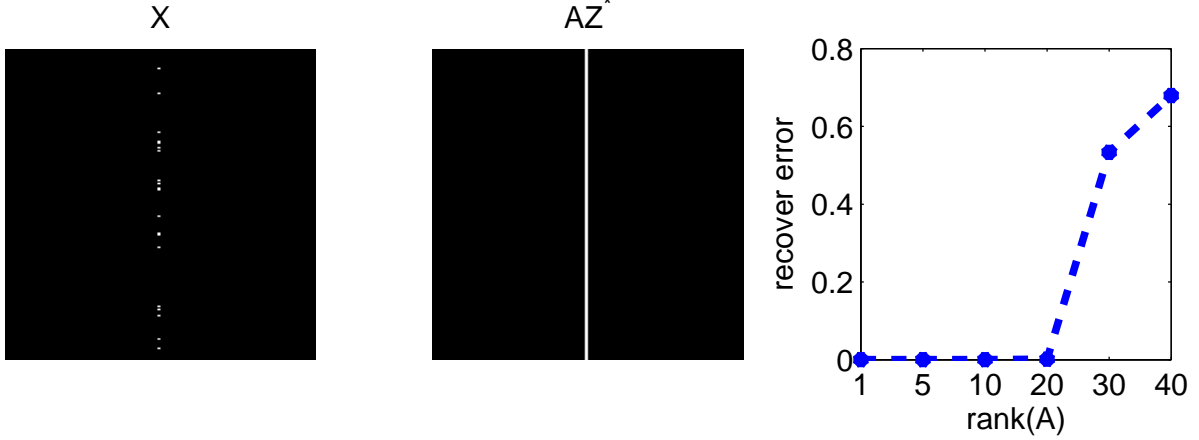


Figure 3: Illustrating that LRFD can avoid μ_2 . In these experiments, L_0 is a 200×200 rank-1 matrix with one column be $\mathbf{1}$ (i.e, a vector of all ones) and everything else being zero. So $\mu_1(L_0) = 1$ and $\mu_2(L_0) = n = 200$. The dictionary in LRFD is set as $A = [\mathbf{1}, W]$, where W is a $200 \times p$ random Gaussian matrix (p is varying). The columns of A are further normalized to have a unit length. As long as $\text{rank}(A) \leq 20$, the latent matrix L_0 (with high coherence) can be exactly recovered from an incomplete observation matrix X , 90% entries of which are missing.

have proven in [3] that, even when the few observed entries are contaminated by a small amount of noise, matrix completion can be accurately performed by the following modified version:

$$\min_L \|L\|_*, \quad \text{s.t.} \quad \|\mathcal{P}_\Omega(X - L)\|_F \leq \epsilon, \quad (5)$$

where $\epsilon > 0$ is a parameter that measures the noise level of the observations.

Similarly, LRFD (2) could be also modified to handle the problem of noisy matrix completion:

$$\min_Z \|Z\|_*, \quad \text{s.t.} \quad \|\mathcal{P}_\Omega(X - AZ)\|_F \leq \epsilon. \quad (6)$$

In the presence of dense noise, it is unrealistic to achieve exact recovery. Yet we have the following theorem to guarantee the recovery accuracy of (6):

Theorem 2 (Noisy) Suppose that the dictionary matrix A with SVD $U_A \Sigma_A V_A^T$ satisfies $\mathcal{P}_{U_A}(U_0) = U_0$ (i.e., $U_0 \subset U_A$), and $\|\mathcal{P}_\Omega(X - L_0)\|_F \leq \epsilon$. For any $\delta > 0$ and some numerical constant $c_a > 0$, if

$$\text{rank}(L_0) \leq \text{rank}(A) \leq \frac{\delta^2 n_2}{c_a \mu_1(A) \log n_1} \quad \text{and} \quad \frac{|\Omega|}{mn} \geq 2\delta,$$

then with probability at least $1 - n_1^{-10}$, the optimal solution (denoted as Z^*) to problem (6) gives a near recovery to L_0 , in a sense that $\|AZ^* - L_0\|_F \leq 2\epsilon/\delta$.

3.3 An Algorithm for Matrix Completion

The theorems introduced above provide a general direction for configuring the dictionary matrix in LRFD, implying several potential procedures. For example, one may drive some kind of optimization framework to jointly compute the variables A and Z . In this paper, we would like to introduce

a simple yet solid algorithm: We firstly obtain an estimate of L_0 by using CONO and then utilize the estimate to construct the dictionary matrix A in LRFD. For the stability of computation, the CONO program (5) is implemented by solving its equivalent version:

$$\min_L \|L\|_* + \frac{\lambda}{2} \|\mathcal{P}_\Omega(X - L)\|_F^2, \quad (7)$$

where $\lambda > 0$ is taken as a parameter. Similarly, our LRFD program (6) is implemented by solving

$$\min_Z \|Z\|_* + \frac{\lambda}{2} \|\mathcal{P}_\Omega(X - AZ)\|_F^2. \quad (8)$$

Provided that the observed entries are contaminated by small Gaussian noise and the dictionary A is column-wisely unit-normed (i.e., $Ae_i = 1, \forall i$), the regularization parameter λ does not require extensive adjustments. Usually, $\lambda = 100$ is a moderately good choice.

Algorithm 1 Matrix Completion

input: An observed data matrix $X \in \mathbb{R}^{m \times n}$, and a support set Ω that stores the locations of the observed entries.

adjustable parameter: λ .

1. Solve for \hat{L}_0 by optimizing (7) with $\lambda = 100$.
2. Estimate the rank of \hat{L}_0 by

$$\hat{r}_0 = \#\{i : \sigma_i > 10^{-3} \sigma_1\},$$

where $\sigma_1 \geq \sigma_2 \cdots$ are the singular values of \hat{L}_0 .

3. Form \tilde{L}_0 by using the rank- \hat{r}_0 approximation of \hat{L}_0 . That is,

$$\tilde{L}_0 = \arg \min_L \|L - \hat{L}_0\|_F^2, \text{ s.t. } \text{rank}(L) \leq \hat{r}_0,$$

which is solved by SVD.

4. Construct a dictionary \hat{A} from \tilde{L}_0 by normalizing the column vectors of \tilde{L}_0 :

$$[\hat{A}]_{:,i} = \frac{[\tilde{L}_0]_{:,i}}{\|[\tilde{L}_0]_{:,i}\|_2}, i = 1, \dots, n,$$

where $[\cdot]_{:,i}$ denotes the i th column of a matrix.

5. Solve for Z^* by optimizing problem (8) with $A = \hat{A}$ and $\lambda = 100$.

output: $\hat{A}Z^*$.

Algorithm 1 summarizes the whole procedure of our algorithm for matrix completion. Note that the post-processing steps (Step 2 and Step 3) that mildly process the solution of CONO is to further encourage low-rank and well-conditioned dictionary, which is a sufficient condition for LRFD to succeed. To facilitate the choice of the parameter λ , Step 4 further normalizes the column vectors and ensure that the produced dictionary is column-wisely unit-normed.

While simple, our Algorithm 1 is guaranteed in theory not to regress backward. That is, whenever CONN has already been successful in recovering L_0 , the claims made in Theorem 1 and Theorem 2 imply that the recovery produced by Algorithm 1 is successful too.

4 Experiments

4.1 Results on Randomly Generated Data

We first verify the effectiveness of our Algorithm 1 on randomly generated matrices. We generate a collection of 200×1000 data matrices according to the model of $X = \mathcal{P}_\Omega(L_0)$: Ω is an index set chosen at random, and L_0 is created by sampling 200 data points from each of 10 randomly generated subspaces. The rank of each subspace varies from 1 to 20 with step size 1, and thus the rank of L_0 varies from 10 to 200 with step size 10. The observation fraction $|\Omega|/(mn)$ varies from 32.5% to 80% with step size 2.5%. For each combination of rank and observation fraction, we run 10 trials, resulting in a total number of 4000 ($20 \times 20 \times 10$) trials.

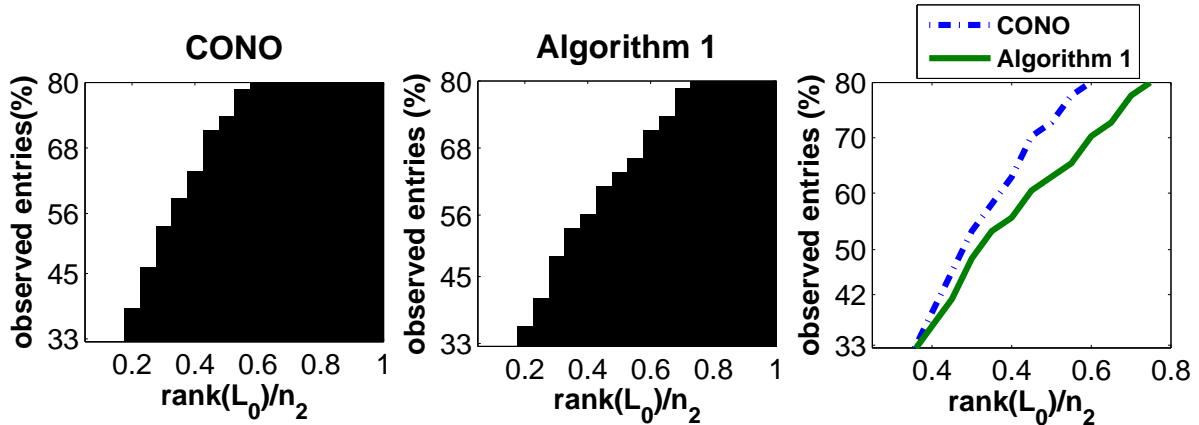


Figure 4: **Comparing CONN with Algorithm 1 on randomly generated matrices.** A curve shown in the third column is the boundary for an algorithm to be successful. In other words, the recovery is successful for any pair $(\text{rank}(L_0)/n_2, |\Omega|/(mn))$ above the curve. Here, the success of recovery is in a sense that $\|\hat{L}_0 - L_0\|_F < 0.05\|L_0\|_F$, where \hat{L}_0 denotes an estimate of L_0 .

Figure 4 compares our Algorithm 1 to CONN, both using $\lambda = 10^6$. It can be seen that the learnt dictionary matrix works distinctly better than the identity matrix adopted by CONN. Namely, the area of the success region (i.e., white region) of our algorithm is 24.6% larger than that of CONN. This verifies the significance of dictionary learning and the effectiveness of our Algorithm 1.

4.2 Results on Motion Data

We now experiment by using real motion sequences with incomplete trajectories. We use 11 additional sequences attached to the Hopkins155 [20] database. Each sequence is a sole dataset (i.e., data matrix) and so there are in total 11 datasets of different properties, including the number of subspaces, the data dimension and the number of data samples. Particularly, in those sequences about 10% of the entries in the data matrix of trajectories are unobserved (i.e., missed) due to vision occlusion, as illustrated in Figure 5.

Notice that the ground truth matrix L_0 is unknown. To evaluate matrix completion algorithms in a quantitative way, we use the *clustering error rates* produced by existing subspace clustering methods as the metrics to evaluate the quality of matrix completion. Namely, we firstly perform subspace clustering on both the incomplete trajectory matrices and the completed versions, and then compute the clustering error rates of the existing subspace clustering methods.

We consider three state-of-the-art subspace clustering methods, including Shape Interaction Matrix (SIM) [11], Low-Rank Representation with *dictionary* = X (LRRx) [12] and Sparse Subspace Clustering (SSC) [21]. As none of these methods owns a mechanism for handling the missing entries, we implement a simple strategy for them: Each missed entry is nominally assigned a value of zero.

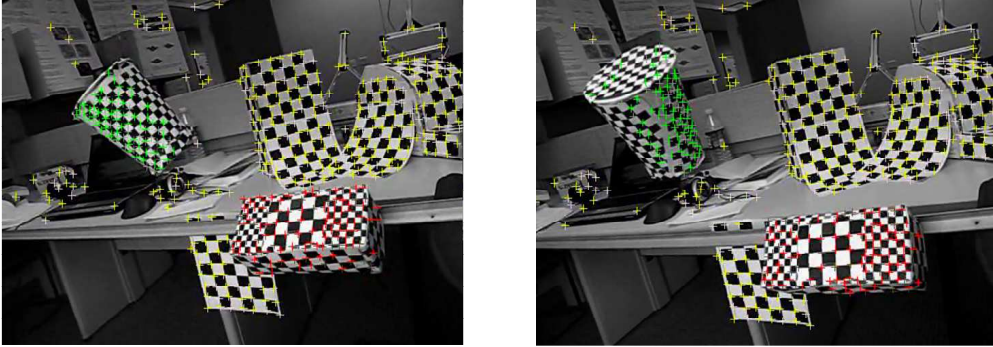


Figure 5: Example image frames from the motion sequences used in our experiments. Due to the rotation of the objects, some measurements in the data matrix of trajectories are missing.

Table 1 shows the error rates of various algorithms. Without the preprocessing of matrix completion, all the subspace clustering methods fail to accurately categorize the trajectories of motion objects, producing error rates higher than 19%. In contrast, without the presence of missing entries, the lowest error rate from SIM, LRRx and SSC on Hopkins155 is as low as 1% [13]. This illustrates that it is important for motion segmentation to restore the missing entries possibly existing in the data matrix of trajectories. By using CONO (with $\lambda = 100$) to restore the missing entries, the clustering performances of all considered subspace clustering methods are improved dramatically. For example, the error rate of SSC is reduced from 31.75% to 3.24%. By seeking an advanced solution for matrix completion using Algorithm 1 (with $\lambda = 100$), the error rates can be reduced again. For example, the error rate of LRRx is reduced from 7% to 5%, which is a 28% improvement. These results verify the effectiveness of our dictionary learning strategy in a realistic environment.

5 Conclusion and Future Work

This paper pointed out that there could exist rich structures inside a low-dimensional subspace, so called as *extra structures beyond low-rankness*. We showed that such extra structures cannot be ignored and have dramatic influences on the success of restoring a low-rank matrix from the incomplete versions. We further proposed a novel model termed LRFD (Low-Rank Factor Decomposition) which handle the extra structures by imposing an additional constraint that the data points are represented by the linear combinations of the bases of a dictionary. Provided that the dictionary is configured properly, LRFD could generally work well on non-uniform data without knowing an precise model of the geometric distributions of the data points. We mathematically proved some theorems which suggest that the dictionary matrix in LRFD should be made low-rank. Subsequently, we established a brief algorithm for approximating such dictionaries in unsupervised environments. Extensive simulations and experiments verify the effectiveness of our algorithm.

The goal of this paper is to analyze in general the problem of modeling extra structures beyond

Table 1: Clustering error rates (%) on the 11 motion sequences with incomplete trajectories.

	SIM	CONN+ SIM	Algorithm 1+ SIM
mean	19.70	12.04	10.76
max	40.04	44.76	38.97
min	3.27	0.58	0.45
std	11.47	15.38	11.79
time (sec.)	0.05	8.42	12.34
	LRRx	CONN+LRRx	Algorithm 1+LRRx
mean	19.85	7.06	4.94
max	36.83	49.68	22.22
min	0.90	0.33	0.33
std	14.66	14.38	6.54
time (sec.)	2.92	9.42	13.03
	SSC	CONN+SSC	Algorithm 1+SSC
mean	31.75	3.24	2.98
max	47.19	12.21	10.28
min	19.25	0	0
std	10.24	4.24	3.87
time (sec.)	2.33	10.47	14.32

low-rankness and provide some basic principles for resolving the problem. Our proposed algorithm does not aim at completely solving the problem, but rather target on a small yet solid step for advancing matrix completion. It is entirely possible to develop more effective algorithms for learning the dictionary matrix in LRFD and we leave this as future work.

Acknowledgement

Guangcan Liu is a Postdoctoral Researcher supported by nsf-dms0808864, nsf-eager1249316, AFOSR-FA9550-13-1-0137, and ONR-N00014-13-1-0764. Ping Li is also partially supported by nsf-iii1360971 and nsf-bigdata1419210.

6 Mathematical Proofs

6.1 Proof of Theorem 1

The same as in CONO, we assume that the locations of the observed entries are selected uniformly at random. In more details, we work with the Bernoulli model $\Omega = \{(i, j) : \delta_{ij} = 1\}$, where δ_{ij} 's are i.i.d variables taking value one with probability $\rho_0 = |\Omega|/(mn)$ and zero with probability $(1 - \rho_0)$, so that the expected cardinality of Ω is $\rho_0 mn$.

We first establish the following lemma that plays an important role in the proof.

Lemma 1 *Suppose $\Omega \sim \text{Ber}(\rho_0)$. Then for any $\delta > 0$, we have*

$$\|\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A}\| \leq 1 - \rho_0 + \delta$$

obeys with probability at least $1 - n_1^{-10}$, provided that

$$\text{rank}(A) \leq \frac{\delta^2 n_2}{c_a \mu_1(A) \log n_1},$$

where c_a is a numerical constant.

Proof For any matrix M , we have

$$\mathcal{P}_{U_A}(M) = \sum_{i,j} \langle \mathcal{P}_{U_A}(M), e_i e_j^T \rangle e_i e_j^T,$$

and so

$$\mathcal{P}_\Omega \mathcal{P}_{U_A}(M) = \sum_{i,j} \delta_{ij} \langle \mathcal{P}_{U_A}(M), e_i e_j^T \rangle e_i e_j^T,$$

which gives

$$\begin{aligned} \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A}(M) &= \sum_{i,j} \delta_{ij} \langle \mathcal{P}_{U_A}(M), e_i e_j^T \rangle \mathcal{P}_{U_A}(e_i e_j^T) \\ &= \sum_{i,j} \delta_{ij} \langle M, \mathcal{P}_{U_A}(e_i e_j^T) \rangle \mathcal{P}_{U_A}(e_i e_j^T). \end{aligned}$$

Note that the Frobenius norm of a matrix is equivalent to the vector ℓ_2 norm, while considering the matrix as a long vector. In that sense, we have

$$\mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A} = \sum_{i,j} \delta_{ij} \mathcal{P}_{U_A}(e_i e_j^T) \otimes \mathcal{P}_{U_A}(e_i e_j^T),$$

where \otimes denotes the Kronecker product.

The definition of $\mu_1(A)$ gives

$$\|\mathcal{P}_{U_A}(e_i e_j^T)\|_F^2 \leq \frac{\mu_1(A) r_A}{m},$$

where $r_A \equiv \text{rank}(A)$. Then by using the results in [18] and following the proof procedure in [1], it could be concluded that the inequality

$$\begin{aligned} \|\rho_0 \mathcal{P}_{U_A} - \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A}\| &\leq \rho_0 (\phi_1 \sqrt{\frac{\mu_1(A) r_A \log n_1}{n_2}} \\ &+ \phi_2 \sqrt{\frac{\mu_1(A) \beta r_A \log n_1}{n_2}}) \leq \phi_1 \sqrt{\frac{\mu_1(A) r_A \log n_1}{n_2}} \\ &+ \phi_2 \sqrt{\frac{\mu_1(A) \beta r_A \log n_1}{n_2}} \end{aligned}$$

obeys with probability at least $1 - n_1^{-\beta}$ for some numerical constants ϕ_1 and ϕ_2 . For any $\delta > 0$, setting $\beta = 10$ and $c_a = (\phi_1 + \sqrt{10}\phi_2)^2$ gives that

$$\|\rho_0 \mathcal{P}_{U_A} - \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A}\| \leq \delta$$

holds with probability at least $1 - n_1^{-10}$, provided that $r_A \leq \delta^2 n_2 / (c_a \mu_1(L_0) \log n_1)$.

By the equality that $\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A} = (1 - \rho_0) \mathcal{P}_{U_A} + (\rho_0 \mathcal{P}_{U_A} - \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A})$ and the triangle inequality,

$$\begin{aligned} \|\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A}\| &\leq \|(1 - \rho_0) \mathcal{P}_{U_A}\| \\ &+ \|\rho_0 \mathcal{P}_{U_A} - \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A}\| \leq 1 - \rho_0 + \delta. \end{aligned}$$

Based on the above lemma, we easily prove the following lemma which states that $(\mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A})^{-1}$ is well defined and has a small operator norm.

Lemma 2 *Let $\|\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A}\| = \psi$. If $\psi < 1$, then the operator $\mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A}$ is an injection from \mathcal{P}_{U_A} to \mathcal{P}_{U_A} , and its inverse operator is given by*

$$\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i.$$

Proof By $\|\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A}\| = \psi < 1$, we have that $\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i$ is well defined and has an operator norm not larger than $1/(1 - \psi)$.

Note that

$$\begin{aligned} \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A} &= \mathcal{P}_{U_A} (\mathcal{I} - \mathcal{P}_{\Omega^\perp}) \mathcal{P}_{U_A} \\ &= \mathcal{P}_{U_A} (\mathcal{I} - \mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A}). \end{aligned}$$

Thus for any $M \in \mathcal{P}_{U_A}$ the following holds:

$$\begin{aligned}
& \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A} (\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i)(M) \\
&= \mathcal{P}_{U_A} (\mathcal{I} - \mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A}) \\
& \quad (\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i)(M) \\
&= \mathcal{P}_{U_A} (\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i - \mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A} \\
& \quad - \sum_{i=2}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i)(M) \\
&= \mathcal{P}_{U_A} (\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i \\
& \quad - \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i)(M) \\
&= \mathcal{P}_{U_A}(M) = M.
\end{aligned}$$

The next lemma finishes to prove Theorem 1.

Lemma 3 *If $\|\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A}\| < 1$, which follows from $|\Omega| > \delta mn$, then $Z^* = A^+ L_0$ is the unique optimal solution to the convex optimization problem (2).*

Proof By $U_0 \subset U_A$, $Z^* = A^+ L_0$ is feasible to (2). By standard convexity arguments [19], $Z^* = A^+ L_0$ is an optimal solution to (2) if there exists a dual vector (or Lagrange multiplier) Y such that

$$A^T \mathcal{P}_\Omega(Y) \in \partial \|A^+ L_0\|_*,$$

where $\partial(\cdot)$ is the sub-gradient of a function. Let the SVD of $A^+ L_0$ be $U \Sigma V^T$. Then we define Y as

$$Y = \mathcal{P}_\Omega \mathcal{P}_{U_A} (\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i) ((A^T)^+ U V^T).$$

With this notation, we have

$$\begin{aligned}
A^T \mathcal{P}_\Omega(Y) &= A^T \mathcal{P}_{U_A} \mathcal{P}_\Omega(Y) \\
&= A^T \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A} \\
& \quad (\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i) ((A^T)^+ U V^T) \\
&= A^T (A^T)^+ U V^T = V_A V_A^T U V^T \\
&= U V^T \in \partial \|A^+ L_0\|_*,
\end{aligned}$$

which gives that $Z = A^+ L_0$ is an optimal solution to the convex optimization problem (2).

It remains to prove that the optimal solution to (2) is unique. We shall consider a feasible perturbation $Z = A^+L_0 + \Delta$ and show that the objective strictly increases whenever $\Delta \neq 0$. By

$$\begin{aligned} 0 &= \mathcal{P}_\Omega(X - A(A^+L_0)) \\ &= \mathcal{P}_\Omega(X - A(A^+L_0 + \Delta)), \end{aligned}$$

we have

$$\mathcal{P}_\Omega(A\Delta) = 0, \quad \text{i.e.,} \quad A\Delta \in \Omega^c.$$

Then, by $A\Delta \in U_A$, we have $A\Delta \in \Omega^c \cap U_A$. This, together with the assumption $\Omega^c \cap U_A = \{0\}$, gives

$$A\Delta = 0, \quad \text{i.e.,} \quad \Delta \in V_A^\perp \subset U^\perp,$$

where $(\cdot)^\perp$ denotes the orthogonal complement of an orthonormal matrix.

We also have

$$\begin{aligned} \|A^+L_0 + \Delta\|_* &= \left\| \begin{bmatrix} U^T \\ (U^\perp)^T \end{bmatrix} (A^+L_0 + \Delta) [V, V^\perp] \right\|_* \\ &= \left\| \begin{bmatrix} U^T A^+L_0 V & 0 \\ (U^\perp)^T \Delta V & (U^\perp)^T \Delta V^\perp \end{bmatrix} \right\|_* \\ &\geq \|U^T A^+L_0 V\|_* \\ &= \|A^+L_0\|_*, \end{aligned}$$

where the equality can hold if and only if

$$(U^\perp)^T \Delta V = 0 \quad \text{and} \quad (U^\perp)^T \Delta V^\perp = 0.$$

This gives $(U^\perp)^T \Delta = 0$, i.e., $\Delta \in U$. However, we have already proven $\Delta \in U^\perp$. Thus, the inequality $\|A^+L_0 + \Delta\|_* > \|A^+L_0\|_*$ strictly holds unless $\Delta = 0$. In other words, $Z^* = A^+L_0$ is the unique optimal solution to (2).

6.2 Proof of Theorem 2

Proof By triangle inequality,

$$\begin{aligned} \|\mathcal{P}_\Omega(AZ^* - L_0)\|_F &= \|\mathcal{P}_\Omega(AZ^* - X) \\ &\quad + \mathcal{P}_\Omega(X - L_0)\|_F \\ &\leq \|\mathcal{P}_\Omega(AZ^* - X)\|_F \\ &\quad + \|\mathcal{P}_\Omega(X - L_0)\|_F \\ &\leq 2\epsilon. \end{aligned}$$

Since $U_0 \subset U_A$, $AZ^* - L_0 \in \mathcal{P}_{U_A}$. By the invertibility of $\mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A}$,

$$\begin{aligned} AZ^* - L_0 &= (\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A})^i) \\ &\quad \mathcal{P}_{U_A} \mathcal{P}_\Omega \mathcal{P}_{U_A} (AZ^* - L_0), \end{aligned}$$

where the validity (with probability at least $1 - n_1^{-10}$) of $\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i$ is from Lemma 2.

It could be calculated that

$$\begin{aligned}
\|AZ^* - L_0\|_F &= \|(\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i) \\
&\quad \mathcal{P}_{U_A} \mathcal{P}_{\Omega} \mathcal{P}_{U_A} (AZ^* - L_0)\|_F \\
&\leq \|(\mathcal{I} + \sum_{i=1}^{\infty} (\mathcal{P}_{U_A} \mathcal{P}_{\Omega^\perp} \mathcal{P}_{U_A})^i)\| \\
&\quad \times \|\mathcal{P}_{U_A} \mathcal{P}_{\Omega} \mathcal{P}_{U_A} (AZ^* - L_0)\|_F \\
&\leq \frac{\|\mathcal{P}_{U_A} \mathcal{P}_{\Omega} \mathcal{P}_{U_A} (AZ^* - L_0)\|_F}{\rho_0 - \delta} \\
&= \frac{\|\mathcal{P}_{U_A} \mathcal{P}_{\Omega} (AZ^* - L_0)\|_F}{\rho_0 - \delta} \\
&\leq \frac{\|\mathcal{P}_{\Omega} (AZ^* - L_0)\|_F}{\rho_0 - \delta} \\
&\leq \frac{2\epsilon}{\rho_0 - \delta} \\
&\leq \frac{2\epsilon}{\delta},
\end{aligned}$$

where the last inequality is concluded from the condition $\rho_0 = |\Omega|/(mn) \geq 2\delta$.

References

- [1] E. Candès and B. Recht, “Exact Matrix Completion via Convex Optimization,” *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.
- [2] C. R. Johnson, “Matrix Completion Problems: A survey,” *Matrix Theory and Applications*, pp. 171–176, 1990.
- [3] E. Candès and Y. Plan, “Matrix Completion with Noise,” *IEEE Proceeding*, vol. 98, pp. 925–936, 2010.
- [4] K. Mohan and M. Fazel, “New Restricted Isometry Results for Noisy Low-Rank Recovery,” *Proc. IEEE International Symposium on Information Theory*, pp. 1573–1577, 2010.
- [5] B. Recht, W. Xu, and B. Hassibi, “Necessary and Sufficient Conditions for Success of the Nuclear Norm Heuristic for Rank Minimization,” *CalTech, Technical Report*, 2008.
- [6] S. Negahban and M. J. Wainwright, “Restricted Strong Convexity and Weighted Matrix Completion: Optimal Bounds with Noise,” *Journal of Machine Learning Research*, vol. 13, pp. 1665–1697, 2012.
- [7] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola, “Cofi Rank - Maximum Margin Matrix Factorization for Collaborative Ranking,” *Proc. Neural Information Processing Systems*, 2007.
- [8] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral Regularization Algorithms for Learning Large Incomplete Matrices,” *Journal of Machine Learning Research*, vol. 11, pp. 2287–2322, 2010.
- [9] M. Fazel, “Matrix Rank Minimization with Applications,” *PhD thesis*, 2002.
- [10] B. Recht, M. Fazel, and P. Parrilo, “Guaranteed Minimum-Rank Solutions of Linear Matrix equations via Nuclear Norm Minimization,” *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.
- [11] J. Costeira and T. Kanade, “A Multibody Factorization Method for Independently Moving Objects,” *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.
- [12] G. Liu, Z. Lin, and Y. Yu, “Robust Subspace Segmentation by Low-Rank Representation,” *Proc. International Conference on Machine Learning*, pp. 663–670, 2010.
- [13] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust Recovery of Subspace Structures by Low-Rank Representation,” *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.
- [14] N. Srebro and T. Jaakkola, “Generalization Error Bounds for Collaborative Prediction with Low-Rank Matrices,” *Proc. Neural Information Processing Systems*, pp. 5–27, 2005.
- [15] F. Wang and P. Li, “Efficient Nonnegative Matrix Factorization with Random Projections,” *Proc. SIAM International Conference on Data Mining*, pp. 281–292, 2010.
- [16] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust Principal Component Analysis?” *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, 2011.
- [17] H. Xu, C. Caramanis, and S. Sanghavi, “Robust PCA via Outlier Pursuit,” *Proc. Neural Information Processing Systems*, 2010.

- [18] M. Rudelson, “Random Vectors in the Isotropic Position,” *Journal of Functional Analysis*, pp. 60–72, 1999.
- [19] R. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.
- [20] R. Tron and R. Vidal, “A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.
- [21] E. Elhamifar and R. Vidal, “Sparse Subspace Clustering,” *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2790–2797, 2009.
- [22] Z. Wang, M. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye “Orthogonal Rank-one Matrix Pursuit for Low Rank Matrix Completion,” *Arxiv*, 2014.
- [23] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor Completion for Estimating Missing Values in Visual Data,” *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.
- [24] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi, “Robust Matrix Completion and Corrupted Columns,” *Proc. International Conference on Machine Learning*, pp. 873–880, 2011.
- [25] Y. Wang and H. Xu, “Stability of Matrix Factorization for Collaborative Filtering,” *Proc. International Conference on Machine Learning*, pp. 417–424, 2012.
- [26] G. Liu and P. Li, “Recovery of Coherent Data via Low-Rank Dictionary Pursuit,” *Arxiv*, 2014.